A study of different families of fusion functions for combining classifiers in the One-vs-One strategy

Mikel Uriz[1], Daniel Paternain[1], Aranzazu Jurio[1], Humberto Bustince[1], and Mikel Galar[1]

Dpto. de Automática y Computación, Universidad Publica de Navarra, Campus Arrosadia s/n, 31006 Pamplona, Spain,
{mikelxabier.uriz, daniel.paternain, aranzazu.jurio, bustince,mikel.galar}@unavarra.es

Abstract. In this work we study the usage of different families of fusion functions for combining classifiers in a multiple classifier system of *One-vs-One* (OVO) classifiers. OVO is a decomposition strategy used to deal with multi-class classification problems, where the original multi-class problem is divided into as many problems as pair of classes. In a multiple classifier system, classifiers coming from different paradigms such as support vector machines, rule induction algorithms or decision trees are combined. In the literature, several works have addressed the usage of classifier selection methods for these kinds of systems, where the best classifier for each pair of classes is selected. In this work, we look at the problem from a different perspective aiming at analyzing the behavior of different families of fusion functions to combine the classifiers. In fact, a multiple classifier system of OVO classifiers can be seen as a multi-expert decision making problem. In this context, for the fusion functions depending on weights or fuzzy measures, we propose to obtain these parameters from data. Backed-up by a thorough experimental analysis we show that the fusion function to be considered is a key factor in the system. Moreover, those based on weights or fuzzy measures can allow one to better model the aggregation problem.

Keywords: Aggregations, Fusion Functions, Classification, One-vs-One, Multiple Classifier System

1 Introduction

In Machine Learning, classification consists in learning a classifier from labeled data capable of assigning the correct label to new patterns. Among classification problems, two different scenarios can be considered depending on the number of classes to be distinguished: binary (two-class) and multi-class problems. Multi-class classification is usually more difficult because the establishment of the decision boundaries become more complex. One possible solution to cope with this difficulty is the usage of decomposition strategies [20], which divide the original multi-class problem into easier to solve binary problems. Evidently, this simplification in the learning phase come at a cost in the combination phase, where the outputs of all the classifiers that were learned for each new sub-problem needs to be combined.

One of the most commonly employed decomposition strategy is *One-vs-One* (OVO). In OVO, as many new sub-problems as possible pairs of classes are created and each one is addressed by an independent base classifier. New instances are classified by being submitted to all the base classifiers, whose outputs are combined. One important advantage of this technique is that it usually performs better even when the underlying classifier is able to address the multi-class problem directly [12].

In this work, we focus on the OVO strategy, and more specifically on the combination phase of Multiple Classifier Systems (MCSs) formed of OVO classifiers. A MCSs is a set formed of classifiers coming from different learning paradigms [17]. In the case of OVO, the idea is that different classifiers may suit better the classification of each pair of classes. For this reason, several previous works have considered the selection of the best classifier for each pair of classes in the MCS [19, 22]. In this work, our aim is to look at this problem as a multi-expert decision making problem, where we have the different experts (types of classifiers) and their preference matrices for the considered alternatives (classes). In this context, we want to study the influence of the fusion function considered to combine the matrices from the different experts into a single one in the classification performance.

In the last decades, the study of aggregation functions has grown significantly, since the necessity of fusing or aggregating quantitative information arises in almost every application [4, 3, 6, 16]. However, in the last years, new extensions of aggregation functions have been proposed, which are able to model the interaction among data in a better way even though classical properties of aggregation functions, such as monotonicity, are not satisfied [21, 23]. From a broad point of view, these extensions are called fusion functions [5].

One of the prominent examples of fusion functions that are able to model the importance of the inputs or the interactions among them is the discrete Choquet integral [8] and its extensions (Choquet-like preaggregation functions) [21], which are based on fuzzy measures. In this work, we propose to construct these measures directly from the knowledge that we can extract from the experts (classifiers) using the training data.

In order to perform this study, we use twenty eight datasets from KEEL [2] and we consider the usage of non-parametric statistical tests to analyze the results obtained [14]. Since we are dealing with multiple classes datasets we will not only consider accuracy measure to evaluate the results, but we will also make use of other measures that give more focus to the correct classification of all classes, such as the average accuracy and the geometric mean. We will develop a hierarchical study, where we consider intra- and inter-family comparisons, to analyze usage of different fusion functions.

The structure of the paper is as follows. In Section 2, recall the different fusion functions considered in this work. Section 3 contains an introduction to the decomposition of multi-class problems, the OVO strategy and the MCSs formed of OVO classifiers. In Section 4, we describe in detail the experimental framework considered for this study, including how to set up the parameters of the parameterizable fusion functions. Section 5 contains the analysis of the results obtained. Finally, in Section 6 we draw the conclusions.

2 Fusion functions

In recent literature, aggregation of quantitative information has been faced by the use of aggregation functions. An aggregation function is defined as a mapping $f:[0,1]^n \to [0,1]$ (the interval $[0,1]$ can be extended to any other interval) such that $f(0,\ldots,0)=0$, $f(1,\ldots,1)=1$ satisfying the monotonicity property, i.e., if $x_i \leq y_i$ for all $i \in \{1,\ldots,n\}$, then $f(x_1,\ldots,x_n) \leq f(y_1,\ldots,y_n)$ [4, 3, 6, 16]. According to [4, 3], the main classes of aggregation functions are the following: averaging, conjunctive, disjunctive and mixed. In this work we mainly (but not only) focus on averaging functions, those which are bounded by the minimum and maximum of inputs.

However, in the last two years the monotonicity property of aggregation functions has been dropped or generalized to new types of monotonicity (see for example [5]). From these studies, new concepts such as preaggregation functions [21] or internal fusion functions [23] have been defined. Since in this paper we model data aggregation from a very broad point of view and we use several non-monotone functions, we have used the more general definition of fusion function (see [5]).

In order to classify the big number of fusion functions considered in this work, we have established a classification based on the necessity of defining weights or measures associated to them. Basically we have considered: unweighted fusion functions, weighted fusion functions and measure-based fusion functions.

Unweighted fusion functions In this subsection we consider classical aggregation functions:

- The arithmetic mean $AM(x_1,\ldots,x_n)=\frac{1}{n}(x_1,\ldots,x_n)$;
- The median $MED(x_1,\ldots,x_n)=\begin{cases}\frac{1}{2}\left(x_{(k)}+x_{(k+1)}\right) & \text{if } n=2k \text{ is even,}\\ x_{(k)} & \text{if } n=2k-1 \text{ is odd,}\end{cases}$ where $x_{(k)}$ stands for the k-th largest (smallest) element of $x_1,\ldots,x_n$;
- The geometric mean $GM(x_1,\ldots,x_n)=(\prod_{i=1}^n x_i)^{\frac{1}{n}}$;
- The harmonic mean $HM(x_1,\ldots,x_n)=n\left(\sum_{i=1}^n \frac{1}{x_i}\right)^{-1}$.

Weighted fusion functions In this subsection we consider fusion functions whose behaviour is modeled by a weighting vector. This means that not every input is equally important for the calculation of the fused value, a fact that clearly allows the incorporation of certain outside information to the fusion process. We will consider a weighting vectors $w=(w_1,\ldots,w_n)$ satisfying $w_i \in [0,1]$ and $\sum_{i=1}^n w_i = 1$ [4, 3].

The weighted fusion functions considered, which in fact are weighted aggregation functions, are:

- The weighted arithmetic mean $WAM(x_1,\ldots,x_n)=\sum_{i=1}^n w_i x_i$;
- The ordered weighted averaging $OWA(x_1,\ldots,x_n)=\sum_{i=1}^n w_i x_{(i)}$, where $(.)$ is a permutation such that $x_{(1)} \geq \cdots \geq x_{(n)}$.

Measure-based fusion functions In this subsection we consider a set of fusion functions that are based on fuzzy measures. Unlike the case of weighted fusion functions, which allow one to model the importance of each individual input, the use of fuzzy measures allows one to model more general interactions among inputs. In this sense,

the importance is given to collections (groups or coalitions) of inputs. Obviously, the construction of the fuzzy measure is the key point for this family of fusion functions.

Definition 1. *Let $\mathscr{N} = \{1,\dots,n\}$. A discrete fuzzy measure is a set function $m : 2^{\mathscr{N}} \to [0,1]$ which is monotonic, i.e., $m(S) \leq m(T)$ whenever $S \subseteq T$ and satisfies $m(\emptyset) = 0$ and $m(\mathscr{N}) = 1$.*

We start mentioning the Choquet integral, which is a prominent example of measure-based averaging operator. We start considering a permutation σ such that $x_{\sigma(1)} \leq \cdots \leq x_{\sigma(n)}$ with the convention $x_{\sigma(0)} = 0$:

– The discrete Choquet integral

$$Ch(x_1,\dots,x_n) = \sum_{i=1}^{n} (x_{\sigma(i)} - x_{\sigma(i-1)}) * m(\{\sigma(i),\dots,\sigma(n)\})$$

As we have mentioned before, in [21] a new type of operator, called pre-aggregation function, was given. One of the easiest ways to construct pre-aggregation is by changing certain operations in the Choquet integral. We have considered the following pre-aggregation functions:

– The Choquet-based operator based on minimum t-norm

$$Ch_M(x_1,\dots,x_n) = \sum_{i=1}^{n} \min\{x_{\sigma(i)} - x_{\sigma(i-1)}, m(\{\sigma(i),\dots,\sigma(n)\})\};$$

– The Choquet-based operator based on Lukasiewicz t-norm

$$Ch_L(x_1,\dots,x_n) = \sum_{i=1}^{n} \max\{0, x_{\sigma(i)} - x_{\sigma(i-1)} + m(\{\sigma(i),\dots,\sigma(n)\} - 1)\};$$

3 One-vs-One decomposition of multi-class problems and multiple classifier systems

In this section we introduce classification problems, and more specifically, the One-vs-One (OVO) strategy to deal with multi-class classification problems and multiple classifier systems aimed at improving classification performance by the combination of several classifiers.

In Machine Learning a classification problem consists in learning a system (classifier) capable of predicting the desired output (label) for each input pattern. Formally, the objective is to find a mapping function $\mathbb{A}^i \to \mathbb{C}$ where $a_1,\dots,a_i \in \mathbb{A}$ are the i features that characterize each input example $x_1,\dots,x_n$ and each input example has associated a desired output $y_j \in \mathbb{C} = \{c_1,\dots,c_m\}$. The classifier is expected to generalize well to examples from the problem that has not been considered in training, that is, it should have a good generalization ability.

A classification problem is said to be a multi-class problem when the number of classes is greater than two ($|\mathbb{C}| > 2$). These problems are considered to be more difficult than binary classification problems since the classification boundaries are usually

more complex and there is a greater overlapping among classes. This is why decomposition strategies [20] came up, to deal with multi-class problems by dividing the original problem into easier to solve binary class classification problems. Therefore, a binary classifier is learned for each new problem, known as base learners, and the outputs of these classifiers are combined when classifying a new unlabeled example. These strategies have proved to be not only useful when working with classifiers that only support binary problems (such as Support Vector Machines, SVMs [25]), but also when considering classifiers with inherent multi-class support. In these cases, the final performance of can also be improved if the problem is decomposed [12].

3.1 The One-vs-One strategy

The OVO strategy is among the most commonly employed decomposition strategies. In this strategy, an m-class problem is divided into as many problems as possible pair of classes, generating $m(m-1)/2$ sub-problems that are faced by independent base classifiers. In each sub-problem, only the examples belonging to a pair of classes are considered, while discarding the rest of them. Then, to classify a new example, it is submitted to all the classifiers whose outputs needs to be combined to decide the final class label. In order to perform the combination, all the outputs are usually stored in a score-matrix (Eq. 1) where each position $r_{ij}, r_{ji} \in [0,1]$ corresponds to the confidence degree of the classifier distinguishing classes $\{C_i, C_j\}$. Since most of the classifiers provide confidence estimates based on probabilities, usually r_{ji} is computed as $r_{ji} = 1 - r_{ij}$. However, if this is not the case, as it occurs with fuzzy rule-based classification systems [10], the score-matrix should be normalized so that $r_{ij} + r_{ji} = 1$ [10].

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \tag{1}$$

Finally, the outputs of the base classifiers are combined for each row (class) and the predicted class label is assigned to the one achieving the greatest total confidence. In the literature, several combination strategies have been developed for this purpose. A thorough review was performed in [12] and several extended combinations have been developed by considering the usage of classifier selection and weighting mechanism [11, 13]. In this work, we consider the Weighted Voting (WV) [18] strategy as it has shown to be a robust yet simple method. In this method, each base classifier votes for both classes based on the confidences provided for the pair of classes. Finally, the class having the largest value is given as output.

$$Class = arg \max_{i=1,\ldots,m} \sum_{1 \leq j \neq i \leq m} r_{ij}. \tag{2}$$

3.2 Combining several OVO in a multiple classifier systems

The OVO strategy can be seen as a ensemble model [12]. Ensembles refer to the combination of classifiers aiming at improving the results of single classifiers. This term

is usually considered to describe the combination of minor variants of the same classifiers. Otherwise, multiple classifier systems (MCSs) is a broader category also including those combinations considering the hybridization of different classification models [17].

Recently, several works have considered the hybridization of OVO ensembles (where the same base classifier is used for each sub-problem, e.g., SVMs) with MCSs. That is, to construct several OVO ensembles with different classifiers (for example, one using SVMs, another using a rule induction method and the other using Decision Trees) and to combine the outputs of all the OVO ensembles to make the final decision.

In previous works, the authors have focused on dynamically or statically selecting the best classifier for distinguishing each pair of classes [19, 22]. However, in this work we aim to look at the problem from a different perspective so as to test the usage of different fusion functions in the combination of the different classifiers.

Once all the OVO classifiers from the MCS have been trained (assuming that we have three different classifiers and a four class problem we would have $3 \cdot 4 \cdot (4-1)/2$ classifiers), a new instance is classified by submitting it to all the classifiers. As a result, instead of obtaining a single score-matrix, we would obtain as many score-matrices as classifiers considered (three in our example). The problem is how to combine these score-matrices into a single one in which we can apply the WV strategy to classify the example. This is why we can understand the problem as a multi-expert decision making problem. Our proposal in this work is to combine the different score-matrices by the usage of fusion function. Our aim is to study how the usage of different fusion functions affects the performance of the MCS. In order to do so, we will consider the different fusion functions reviewed in the previous section and we will propose different mechanism to assign the weights or create the fuzzy measures in the functions requiring these parameters. More details on how these parameters are obtained are given in Section 4.2

4 Experimental framework

4.1 Datasets, performance measures, statistical tests and algorithms

In order to carry out the experimental study, we use twenty-eight numerical datasets selected from the KEEL dataset repository [2], whose main features are introduced in Table 1.

The result for each method and dataset is obtained using a 5 fold cross-validation scheme. Moreover, in order to properly analyze the results obtained, we have applied non-parametric statistical tests [14]. More specifically, we use the Wilcoxon test to compare a pair of methods, whereas the Friedman aligned ranks test is considered to compare a group of methods in order to detect whether statistical differences exist. In such a case, the Holm *post-hoc* test is performed to find the algorithms that reject the null hypothesis of equivalence against the selected control method.

Given that we are dealing with multi-class problems, we have considered three different performance measures to analyze the results obtained: Accuracy rate (Acc), that is, the ratio of correctly classified examples; Average Accuracy Rate (AvgAcc), which refers to the average of the ratio of correctly classified examples per class; Geometric Mean (GM), the geometric mean of the ratio of correctly classified examples per class.

Table 1. Summary of the features of the datasets used in the experimental study.

Dataset	#Ex.	#Atr.	#Clas.	Dataset	#Ex.	#Atr.	#Clas.
autos	159	25	6	nursery	1296	8	5
balance	625	4	3	pageblocks	548	10	5
car	1728	6	4	penbased	1100	16	10
cleveland	297	13	5	satimage	643	36	7
contraceptive	1473	9	3	segment	2310	19	7
dermatology	358	34	6	shuttle	2175	9	7
ecoli	336	7	8	splice	319	60	3
flare	1066	11	6	tae	151	5	3
glass	214	9	7	thyroid	720	21	3
hayes-roth	132	4	3	vehicle	846	18	4
iris	150	4	3	vowel	990	13	11
led7digit	500	7	10	wine	178	13	3
lymphography	148	18	4	yeast	1484	8	10
newthyroid	215	5	3	zoo	101	16	7

Hence, Acc gives us a global measure of quality of the algorithm, whereas AvgAcc and GM are more focused on properly measuring whether all the classes of the problem are being properly classified or not (being the GM much more restrictive than AvgAcc).

Regarding the classification algorithms considered to form our MCS of OVO classifiers, we have considered the following ones (which were also considered in our previous works on the topic [12, 11, 13]): *Support Vector Machine* (SVM) [25], *C4.5 decision tree* [24], *k−Nearest Neighbors* (*k*NN) [1], *Repeated Incremental Pruning to Produce Error Reduction* (Ripper) [9], *Positive Definite Fuzzy Classifier* (PDFC)[7].

These classifiers were trained using the parameters shown in Table 2. These values are common for all problems, and they were selected according to the recommendation of the corresponding authors, which is also the default setting of the parameters included in KEEL[1] software [2] used to develop our experiments. We treat nominal attributes in SVM and PDFC as scalars to fit the data into the systems using a polynomial kernel.

Table 2. Parameter specification for the base learners employed in the experimentation.

Algorithm	*Parameters*
SVM_{Poly}	C = 1.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12, Kernel Type = Polynomial, Polynomial Degree = 1 Fit Logistic Models = True
SVM_{Puk}	C = 100.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12, Kernel Type = Puk, PukKernel ω = 1.0, PukKernel σ = 1.0 Fit Logistic Models = True
C4.5	Prune = True, Confidence level = 0.2, Minimum number of item-sets per leaf = 2
3NN	$k = 3$, Distance metric = HVDM
Ripper	Size of growing subset = 66%, Repetitions of the optimization stage = 2
PDFC	C = 100.0, Tolerance Parameter = 0.001, Epsilon = 1.0E-12, Kernel Type = Polynomial, Polynomial Degree = 1, PDRF Type = Gaussian

We should notice that score-matrices should store the confidences obtained from the classifiers. Since not all the classifiers provide confidences straightforwardly, we detail how they have been obtained hereafter.

– **SVM –** Probability estimates from the SVM.

[1] `http://www.keel.es`

- **C4.5 –** Accuracy of the leaf making the prediction (correctly classified train examples divided by the total number of covered train instances).
- ***k*NN –** Distance-based confidence estimation. $Confidence = \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}}$ where d_l is the distance between the input pattern and the l^{th} neighbor and $e_l = 1$ if the neighbor l is from the class and 0 otherwise.
- **Ripper –** Accuracy of the rule used in the prediction (computed as in C4.5 considering rules instead of leafs).
- **PDFC –** The prediction of the classifier, that is, confidence equal to 1 is given for the predicted class.

4.2 Estimation of the parameters for the fusion functions

Hereafter, we present the way in which the parameters required for some of the fusion functions are estimated.

Weight calculation For the weighted arithmetic mean we need to set the weights for each input (classifier, e.g., SVM, 3NN, ...). We set each weight as the normalized accuracy of each method in the training dataset, that is, $w_i = \frac{Acc_i}{\sum_{j=1}^{n} Acc_j}$ for all $i \in \{1,\ldots,n\}$.

Moreover, we have used two different versions for weighted fusion functions: a global and a local approach. In the global approach, we set one weight per classifier. However, in the local approach, each classifier gets a weight for each individual problem (accuracy over the pair of classes).

The calculation of the weights for OWA operators is done by means of increasing fuzzy quantifiers (see [26]), which are given by $w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right)$ for all $i \in \{1,\ldots,n\}$. In this work we have considered 3 different fuzzy quantifiers yielding three OWA operators : 'at least half' (OWA_alh) with $a = 0, b = 0.5$; 'as many as possible' (OWA_amap) with $a = 0.5, b = 1$; and 'most of them' (OWA_mot) with $a = 0.3, b = 0.8$.

Fuzzy measure values For the measure-based fusion functions, we need to build a fuzzy measure $m : 2^{\mathscr{N}} \to [0,1]$ with $\mathscr{N} = \{1,\ldots,n\}$, being n the number of classifiers considered. We will start by considering the uniform fuzzy measure m_U which is given by $m_U(A) = \frac{|A|}{n}$ for every $A \subseteq \mathscr{N}$. It is clear that the Choquet integral with respect to a uniform measure is nothing but the arithmetic mean.

However, in order to capture the interactions among classifiers by means of the fuzzy measure, we will take the individual accuracy of each classifier as well as the accuracy of each possible combination of classifiers. We will denote these accuracies as Acc_A, for all $A \subseteq \mathscr{N}$. Now, for each level of the fuzzy measure (all the elements of the fuzzy measure with the same cardinality), we calculate the arithmetic mean of accuracies in the corresponding level, namely $MeanAcc_i$ for every $i \in \{1,\ldots,n\}$. Finally, the value of the fuzzy measure for each $A \subseteq \mathscr{N}$ will be given by

$$m(A) = m_U(A)(1 + Acc_A - MeanAcc_{|A|}). \tag{3}$$

Taking this expression into account, the accuracies of classifiers that are better than the average accuracy in the same level will be increased and those that are worse will be decreased with respect to the uniform measure. In a similar way as in the previous

calculation of weights, we will consider a global and a local approach for each measure-based fusion functions.

Notice that we cannot guarantee the monotonicity of m for every possible value of Acc and $MeanAcc$. To correct it, and based on the monotonicity verification given in [15], we use a top-down monotonicity correction: we start from the top level of the measure ($m(\mathscr{N})$ and we evaluate the measure values of the level above ($m(A)$ where $|A| = n-1$). If we find some A such that $m(A) > m(\mathscr{N})$, then we set $m(A) = m(\mathscr{N})$. Once the $n-1$-th level is verified (w.r.t the n-th level), we check the $n-2$-th level w.r.t. the $n-1$-th level. We repeat the procedure until the whole measure satisfies the monotonicity criterion.

5 Experimental study

On the one hand, Table 3 shows the accuracy (Acc), the average accuracy per class (AvgAcc) and the geometric mean of each class accuracy (GM) obtained in testing using the different fusion functions to combine the OVO score-matrices in the MCS. The best result in each performance measure is underlined

Table 3. Average test results over all datasets obtained with the different fusion functions for each performance measure

Family	Fusion	Acc	AvgAcc	GM
Unweighted	AM	0.8544	0.7911	0.6240
	MED	0.8580	0.7951	0.6332
	GM	0.8285	0.7535	0.5588
	HM	0.8252	0.7515	0.5610
Weighted	WAM	0.8544	0.7916	0.6308
	WAM_local	0.8481	0.7893	0.6344
	OWA_alh	0.8573	0.7996	0.6448
	OWA_amap	0.8496	0.7815	0.6073
	OWA_mot	0.8554	0.7921	0.6254
Choquet	Ch	0.8552	0.7940	0.6305
	Ch_local	0.8541	0.7924	0.6334
	Ch_L	0.8487	0.7789	0.6087
	Ch_L_local	0.8502	0.7803	0.6088
	Ch_M	0.8548	0.7939	0.6395
	Ch_M_local	0.8556	0.7964	0.6397

On the other hand, Figure 1 summarizes the statistical study carried out for each performance measure in order to analyze which is the best performer fusion function in each case. In order to create this figure, for each performance measure, we have confronted the functions in each family following Friedman Aligned ranks test. Then, the best performers of each family are compared in the final stage that gives us the best fusion function. In each comparison, we show the ranks obtained by each method (the lower the better) and we remark in **bold-face** the ranks when the post-hoc test shows that there exist significant differences (with $\alpha = 0.1$) in favor of the winning method.

Finally, we have completed our statistical analysis by comparing the arithmetic mean (AM, which the most commonly considered function) with the winner of each

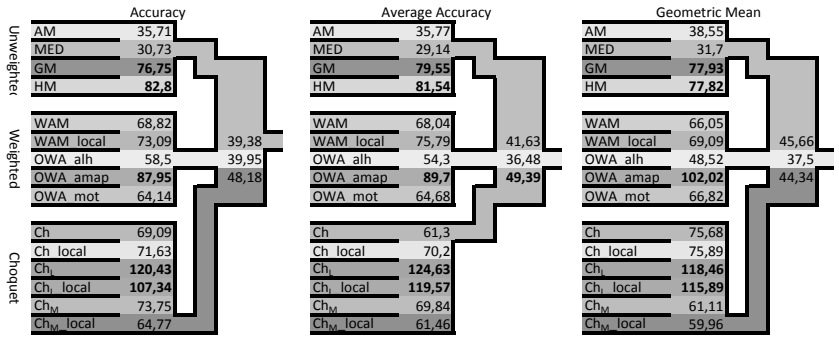

Fig. 1. Hierarchical statistical study comparing the fusion functions in each family and the best performers of each family for each performance measure using Friedman Aligned ranks test.

intra-family comparison. These comparisons are presented in Table 4, where the p-values obtained for each comparison between AM and the corresponding fusion function are presented. Statistically significant differences are presented in **bold-face**

Table 4. Wilcoxon's tests comparing AM vs the best fusion function in each performance measure.

Perf. Measure	Unweighted	Weighted	Choquet
Acc	MED	OWA_alh	Ch_M_local
	0.0152	**0.0298**	0.7610
AvgAcc	MED	OWA_alh	Ch
	0.0194	**0.0126**	**0.0994**
GM	MED	OWA_alh	Ch_M_local
	0.0169	**0.0036**	**0.0400**

Attending at these results, we can observe the following facts.

– Analyzing the results for each family, first, among unweighted functions AM and MED are the best performing ones. Interestingly, MED is statistically outperforming AM following the Wilcoxon test in all the three performance measures. Looking at weighted functions it is interesting to note that OWA_alh is the best performing one, even though statistical differences only exist with respect to OWA_amap. This is possibly due to the fact that the corresponding weighting function acts as an average of the three most competitive classifiers. In this case, obtaining the weights from data (WAM and its local version) has result in worse results than establishing a predefined weights. Finally, regarding fuzzy measure-based functions, pre-aggregations considering the minimum are constantly the best in almost all cases, showing its robustness independently of the measure considered (although no statistical differences are found).

One would expect better performance in the cases where the parameters have been obtained from data, i.e., weighted and measure-based functions. Even though no significant differences are found with respect to WAM and Choquet, in the future our aim is to focus on this functions and try to better model the parameters in order to make the more competitive. In fact, Choquet can recover any OWA operator and hence, intuitively, one should be able to obtain a fuzzy measure leading to at least the same behavior as any OWA (and probably better).

– Finally, looking at Table 4 one can observe that the most commonly considered fusion function in ensembles and MCSs need not be the performing one. AM is statistically outperformed by MED and OWA_alh in all cases and by Choquet in the cases of AvgAcc and GM. Hence, there is margin for improvement by considering different fusion functions

6 Conclusions

In this work, we have considered an MCSs formed of OVO classifiers and looked at the combination phase as a multi-expert decision making problem. Consequently, we have developed a thorough empirical study in order to analyze the behavior of different families of fusion functions. We have also proposed different ways to obtain the parameters of weighted and fuzzy measure-based fusion functions from data. Even though one could expect better performance from these kind of fusion functions, OWAs with specific weights are the ones with the best results. Since OWAs are a particular case of some fuzzy measure-based functions, this fact encourages us to further study different ways of building the fuzzy measures in order to improve the quality of their results.

Acknowledgments. This work was supported in part by the Spanish Ministry of Science and Technology under Project TIN2016-77356-P (AEI/FEDER, UE).

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)
2. Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3, 255–287 (2011)
3. Beliakov, G., Bustince, H., Pradera, A.: A Practical Guide to Averaging Functions. Springer, 2nd edn. (2015)
4. Beliakov, G., Pradera, A., Calvo, T.: Aggregation Functions: A Guide for Practitioners. Springer (2007)
5. Bustince, H., Fernandez, J., Kolesárová, A., Mesiar, R.: Directional monotonicity of fusion functions. European Journal of Operational Research 244, 300–308 (2015)
6. Calvo, T., Mayor, G., Mesiar, R.: Aggregation Operators. New Trends and Applications. Physica-Verlag (2002)
7. Chen, Y., Wang, J.Z.: Support vector learning for fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 11(6), 716–728 (2003)

8. Choquet, G.: Theory of capacities. Ann. Inst. Fourier 5, 1953–1954 (1953)
9. Cohen, W.W.: Fast effective rule induction. In: ICML'95: Proc. of the Twelfth Int. Conf. on Machine Learning. pp. 1–10 (1995)
10. Elkano, M., Galar, M., Sanz, J., Fernandez, A., Barrenechea, E., Herrera, F., Bustince, H.: Enhancing multi-class classification in farc-hd fuzzy classifier: On the synergy between n-dimensional overlap functions and decomposition strategies. IEEE Transactions on Fuzzy Systems 23(5), 1562 – 1580 (2015)
11. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: Dynamic classifier selection for one-vs-one strategy: Avoiding non-competent classifiers. Pattern Recognition 46(12), 3412–3424 (2013)
12. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition 44(8), 1761 – 1776 (2011)
13. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: DRCW-OVO: Distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems. Pattern Recognition 48(1), 28–42 (2015)
14. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180, 2044–2064 (2010)
15. Grabisch, M.: A new algorithm for identifying fuzzy measures and its application to pattern recognition. In: Int. Joint Conf. of the 4th IEEE Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symposium. pp. 145–150 (1995)
16. Grabisch, M., Marichal, J.L., Mesiar, R., Pap, E.: Aggregation Functions. Cambridge University Press (2009)
17. Ho, T.K., Hull, J.J., Srihari, S.N.: Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(1), 66–75 (1994)
18. Hüllermeier, E., Vanderlooy, S.: Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. Pattern Recognition 43(1), 128–142 (2010)
19. Kang, S., Cho, S., Kang, P.: Multi-class classification via heterogeneous ensemble of one-class classifiers. Engineering Applications of Artificial Intelligence 43, 35–43 (2015)
20. Lorena, A.C., Carvalho, A.C., Gama, J.M.: A review on the combination of binary classifiers in multiclass problems. Artificial Intelligence Review 30(1-4), 19–37 (2008)
21. Lucca, G., Sanz, J., Dimuro, G., Bedregal, B., Mesiar, R., Kolesárová, A., Bustince, H.: Preaggregation functions: Construction and an application. IEEE Transactions on Fuzzy Systems 24, 260–272 (2016)
22. Mendialdua, I., Martnez-Otzeta, J.M., Rodriguez-Rodriguez, I., Ruiz-Vazquez, T., Sierra, B.: Dynamic selection of the best base classifier in One versus One. Knowledge-Based Systems 85, 298–306 (2015)
23. Paternain, D., Campión, M.J., Bustince, H., Perfilieva, I., Mesiar, R.: Internal fusion functions. IEEE Transactions on Fuzzy Systems (InPress)
24. Quinlan, J.R.: C4.5: Programs for Machine Learning. San Mateo-California: Morgan Kaufmann Publishers, 1st edn. (1993)
25. Vapnik, V.: Statistical Learning Theory. New York: Wiley (1998)
26. Yager, R.: Quantifier guided aggregation using owa operators. International Journal of Intelligent Systems 11, 49–73 (1998)